REMARKS/ARGUMENTS

Claims 1-24 were pending. Claims 13-24 were regarded as withdrawn. Claims 1-12 were rejected under 35 U.S.C. 102 or 103, with the primary reference being U.S. Patent 6,473,433 to Bianchini, Jr. et al. ("Bianchini").

Herein, claims 13-24 are cancelled. Claims 1, 2, 7 and 11 are amended to correct obvious errors or to clarify what was implicit in the claims.

An Information Disclosure Statement is submitted herewith.

Claim 1 includes the feature: "a level monitor at a receiving blade that monitors levels of the data received and stored at the receiving blade." Bianchini lacks such a "level monitor." The Examiner cites to the "memory controller" of Bianchini as having this feature, which it doesn't, but in any event, Bianchini's memory controllers are each in a respective switch fabric, not at the receiving blade. See Bianchini, Fig. 1. Accordingly, claim 1 is allowable.

Bianchini also lacks claim 1's feature of "a stripe synchronization error detector that detects a stripe synchronization error based on the amount of stored data monitored by said level monitor." The Examiner points to Bianchini's Unstriper as performing this function, but the Unstriper does no such thing. The Unstriper detects errors in data flows across the switch fabric by use of a checkword appended to the data stream by the Striper. This is related to the N+1 redundancy of Bianchini's fabric. Col. 6, line 1 et seq. The switch has, in the example, three fabrics and a spare fabric (total four fabrics). A checkword is generated and appended to the data stream by the Striper, which then stripes the data stream across the 4 fabrics. The Unstriper reconstructs 4 "tentative" data streams from the 4 stripes, and then checks the accuracy of the tentative data streams using the checkword. If there is a match between the checkword appended by the Striper and the checkword that the Unstriper calculated for each of the 4 tentative data

streams, then the fabric operated correctly. If not, and assuming that only one of the four fabrics failed (col. 6, line 32-33), then the Unstriper can determine which of the 4 fabrics failed, can identify the correct data stream out of the four tentative data streams based on a checkword match, and can forward the correct data stream "off chip," i.e., downstream (see Fig. 2 or Fig. 3). "Off chip" refers to the fact that the Unstriper is an ASIC as shown in Figs. 2 and 3.

Accordingly, the Examiner is incorrect in his assertion that Bianchini's Unstriper meets claim 1's feature of "a stripe synchronization error detector that detects a stripe synchronization error based on the amount of stored data monitored by said level monitor." The Unstriper does not detect errors "based on the amount of stored data monitored by said level monitor."

Regarding claim 2, Bianchini lacks the feature of "said level monitor monitors the levels of data stored in each data structure of the receiving blade." Again, the "memory controller" of Bianchini is in Bianchini's switch fabric (Fig. 1), and therefore cannot be the "level monitor" of claim 1 or 2, which monitors levels of data stored at the "receiving blade".

Further, Bianchini lacks claim 2's feature of "said stripe synchronization error detector detects at least one of an overflow and underflow condition in the amount of stored data received on a respective stripe from a particular source." Again, the Examiner's rejection is based on the discussion of the Unstriper's role in the N+1 redundancy of Bianchini's fabric. Col. 6, line 1 et seq. The Examiner's statement at page 3 that "the Unstriper detects that one of the data streams is forwarded off chip and the stream is the faulty stripe and thus an underflow condition occurs" does not accurately reflect Bianchini's operation. Bianchini's goal is to forward the data stream "off chip," i.e., downstream, because the Unstriper is an ASIC. The Examiner incorrectly interpreted this as an erroneous underflow condition. Accordingly, claim 2 is allowable, because Bianchini does not work the way the Examiner describes.

Claims 3-6 are dependent and also allowable.

Regarding claim 7, the rejection of the claim over Bianchini is erroneous for reasons similar to those stated for claims 1 and 2. For instance, Bianchini lacks elements (c) and (d) of claim 7, which state: "(c) monitoring the levels of data stored in each said data structure; and (d) detecting at least one of an overflow and underflow condition in the amount of stored data received on a respective stripe from a particular source." The "data" of element (c) already has been received, sorted, and stored at the receiving blade per elements (a) and (b) of claim 7. Since the memory controller of Bianchini is in the switch fabric rather than at the receiving blade, Bianchini cannot not show element (c) of claim 7. Further, the "off-chip" underflow condition that the Examiner postulates for element (d) of claim 7 is based on a misunderstanding of Bianchini's teaching, as described above for claims 1 and 2. Accordingly, element (d) of claim 7 is missing from Bianchini. Bianchini does not work the way the Examiner describes. Accordingly, claim 7 is allowable.

Claims 8 and 9 depend on claim 7, and are allowable for at least the same reasons as claim 7.

Regarding claim 10, the rejection is also submitted to be erroneous. Claim 10 includes the feature "sending a common character in striped cells in all lanes for a predetermined number of cycles." The Examiner cites to the "resynch cell" that Bianchini describes at col. 18, line 6 et seq. Bianchini states: "A resynch cell is broadcast to all fabrics (new and existing)." This does not meet claim 10. There is no showing of "in striped cell<u>s</u>" or "in all lanes" or "for a predetermined number of cycle<u>s</u>."

Further, Bianchini does not include claim 10's feature of "(c) detecting when an in-synch condition is present that indicates the stripe receive synchronization queues have been cleared."

There is no such detection in Bianchini. According to Bianchini, "[a] queue resynch ends when one of two events happens: 1. A timer expires. 2. The amount of new traffic (traffic received after the resynch cell) exceeds a threshold." Col. 18, line 11 et seq. Neither of these events indicates, directly, that the "queues have been cleared," as is claimed. Both are unrelated to the actual clearing of the queues. The time selected for the timer to expire is merely a guess as to when the queue might clear, and the amount of new traffic is a function of what the switch receives from the outside world. Accordingly, claim 10 is allowable.

Regarding claim 11, Bianchini lacks the feature of "monitoring the level of stripe receive synchronization queues, the stripe receive synchronization queues storing data that passed through the switching fabric." The memory controller of Bianchini queues and dequeues, but cannot perform the monitoring step of claim 11 because the memory controller is part of Bianchini's switching fabric, and the claim refers to data that passed through the switching fabric already. Accordingly, claim 11 is allowable.

Claim 12 includes elements similar to claim 10, and is allowable for the reasons stated above.

CONCLUSION

Accordingly, claims 1-12 should be allowed. Please consider the Information Disclosure Statement mentioned above, and if the case is in a condition for allowance, please issue a Notice of Allowance. Please direct any questions or comments to the undersigned at 408 207 1323.

Respectfully submitted,

Dated: January 16, 2007

James E. Parsons
Reg. No. 34,691

Customer Number: 33,707